

# Active Rare Class Discovery and Classification using Dirichlet Processes

Tom S. F. Haines · Tao Xiang

Received: date / Accepted: date

**Abstract** Classification is used to solve countless problems. Many real world problems, such as visual surveillance, contain uninteresting but common classes alongside interesting but rare classes. The rare classes are often unknown, and need to be discovered whilst training a classifier. Given a data set active learning selects the members within it to be labelled for the purpose of constructing a classifier, optimising the choice to get the best classifier for the least amount of effort. We propose an active learning method for scenarios with unknown, rare classes. By assuming a non-parametric prior on the data the goals of new class discovery and classification refinement are automatically balanced, without any tunable parameters. The ability to work with any specific classifier is maintained, so it may be used with the technique most appropriate for the problem at hand. Results are provided for a large variety of problems, demonstrating superior performance.

**Keywords** Active Learning · Rare Class Discovery · Classification

## 1 Introduction

Classification is an important technique, key to solving innumerable problems. A training set is collected, and a domain expert labels each exemplar in the set with the desired (discrete) answer. The relationship between

the exemplars and the labels is then learned by a classification algorithm, such that the answer can be estimated for future exemplars. As domain experts are not cheap the greatest expense often lies in the labelling step. In many real-world problems, such as visual surveillance, computer network intrusion detection and financial fraud pattern detection, the proportion of exemplars in different classes is imbalanced - the majority belong to uninteresting background classes whilst the interesting classes have few exemplars. This imbalance can dramatically increase the labelling cost, as many more exemplars have to be labelled by the domain expert to have a reasonable chance of including all the rare classes. Furthermore, the interesting minority is often unknown in advance. For example in the Sloan Digital Sky Survey most of the survey images of galaxies and quasars capture known phenomena, whilst unusual phenomena that could be evidence of new science constitute only 0.001% of the total data set (Pelleg and Moore, 2004). To classify rare classes one typically needs to exhaustively label a sizable data set, to obtain sufficient instances of each rare class. Such a manual labelling process is often prohibitively expensive, rendering supervised learning impractical.

Active learning (Settles, 2009) offers a solution. It selects the exemplars to be labelled, with the choice made to minimise the number of labels required to train a good classifier. Because we have unknown classes that are also rare there are two competing goals to consider - to find all the rare classes, and to refine the boundaries between the currently known classes. Both of these behaviours will improve classification performance - if a class is unknown then the classifier will incorrectly classify all instances of that class, whilst boundary refinement is needed to get good performance for classes that have already been discovered. However, most ex-

---

T.S.F. Haines  
Queen Mary, University of London, Mile End Road, London  
E1 4NS, UK  
E-mail: thaines@eecs.qmul.ac.uk

T. Xiang  
E-mail: txiang@eecs.qmul.ac.uk

isting active learning methods either assume that all classes are known and thus focus on the classification problem (Settles, 2009), or focus on the class discovery problem only (Pelleg and Moore, 2004; He and Carbonell, 2007; Vatturi and Wong, 2009). The approaches that try to meet both goals simultaneously (Hospedales et al., 2011; Stokes et al., 2008) are heuristic, and have free parameters that need tuning for each scenario. It should be noted that discovering rare classes is often less of an improvement for classification performance than refining the boundary between common classes. Consequentially it does not make sense to first perform discovery then boundary refinement, particularly as it is impossible to know when all classes have been found. Instead the two goals have to be considered simultaneously and queries made accordingly.

We propose a novel active learning approach, which automatically balances the two competing goals, without the need to tune parameters. It is a pool based approach - it iteratively selects an exemplar from a pool and asks the user to label it. During each iteration the model is updated with the new label, so it can be used to select the next exemplar from the pool. The selection proceeds in three steps. Firstly, for each exemplar in the pool the probability of it belonging to each existing class, and belonging to a new class, are calculated, under a Dirichlet process (DP) assumption. Secondly, the probability that the instance will be misclassified is calculated. Misclassification probability is an *uncertainty* based method, that works to improve the boundary between existing classes (Settles, 2009); however, because a DP assumption allows the probability of belonging to an *unknown* class to be factored in, it also achieves the goal of class discovery. The balance between the two goals is determined by the concentration parameter of the DP, which is automatically inferred. Finally, a single instance is selected, based on the estimated chances of misclassification. Our key contribution is this novel active learning criterion, which is specifically designed to balance the two competing goals of discovery and classification. Furthermore, its implementation is simple, it has no tunable parameters and it works with any probabilistic classifier<sup>1</sup>.

In the following section the relationship between this work and others is explored. Section 3 details the actual algorithm, after which it is evaluated in section 4. Finally, conclusions are given in section 5.

## 2 Related Work

Active learning is a long standing (Angluin, 1988) and expansive field - the survey of Settles (2009) gives an overview, whilst Olsson (2009) also gives a literature review, but focused on natural language processing alone. Such techniques can be broken down into two parts - a learning algorithm and an active learning criterion, which are often integrated and then targeted at a specific problem domain. The criterion is responsible for determining which exemplars are to be labelled. As we are proposing a domain-agnostic and learner-agnostic approach it is the criterion that will now be considered - there are only a few specific approaches, as most papers are about adapting an approach to a specific domain or learning technique.

Random sampling<sup>2</sup> is the simplest possible criterion, where exemplars are selected at random to be labelled. Despite its simplicity when dealing with balanced data the odds are that each random item is a new class, and it will often do surprisingly well. It is inappropriate for unbalanced data however, as the odds of selecting a rare class can become arbitrarily small.

### 2.1 Query by uncertainty

Uncertainty criteria (Lewis and Gale, 1994) select instances for which the classifier is uncertain - they are thus good at refining the boundaries between classes. Multiple uncertainty methods exist (Settles, 2009): One commonly used technique is based on entropy (Settles, 2009) - the entropy of the class membership distribution for each exemplar is calculated, and the highest scoring selected. A high entropy indicates a lot of uncertainty in the classification of an exemplar, with a particular emphasis on exemplars for which many classes are considered to be a reasonable classification possibility. Lewis and Gale (1994) deal with binary classification, and explicitly select the exemplar with membership probability closest to 0.5, the class boundary. Culotta and McCallum (2005) generalise this idea to multiple classes, by selecting the exemplar for which the classifier is least confident. This is related to our approach as it is optimising the same goal - they both select the exemplar that is most likely to be misclassified, but it only considers known classes, hence it can only improve class boundaries.

<sup>1</sup> An implementation is available from <http://thaines.com>

<sup>2</sup> Sometimes referred to as *passive learning*.

## 2.2 Query by committee

Query by committee (QBC) (Seung et al., 1992) requires the existence of multiple classifiers for the labelled data, and consists of selecting exemplars based on a measure of disagreement between them. Obtaining multiple classifiers can be explicit, or it can involve a probability distribution over the classifier, from which multiple specific classifiers can be drawn. Classifiers that include a random element can provide this capability, e.g. boosting and bagging Abe and Mamitsuka (1998). Taking the probabilistic interpretation the committee members can be integrated out, using, for instance, an uncertainty-based metric, but this does not utilise disagreement. A common measure of disagreement is to let each member vote for the exemplars class, treat this as a probability distribution and calculate the entropy (Dagan and Engelson, 1995). McCallum and Nigam (1998) provide an alternative, where they sum the Kullback-Leibler divergence between each committee member’s probabilistic assignment and the consensus of the committee, calculated by averaging. Query by committee works because it considers the space of classifiers that fit the data, and selects exemplars to maximally reduce that space, to get the best classifier quickly. This tends to focus on outliers however, at the expense of boundary refinement.

## 2.3 Expected error reduction

Expected error reduction (Roy and McCallum, 2001) selects the exemplar from the pool that will minimise an estimate of future error. For each exemplar it considers the model after it has been updated with each possible label, estimating the error of each model using all exemplars. This includes those for which the class is known, for which it is a direct comparison, but for exemplars still in the pool it uses the probabilistic labelling of the current model. The expected reduction is then calculated for each exemplar in the pool, and the exemplar with the largest chosen. Whilst arguably the best approach for boundary refinement, it does not do class discovery, and is computationally expensive, so much so that sampling and efficient incremental learning techniques are required (Roy and McCallum, 2001). The approximation of expected error for a future model state is problematic, and can result in it selecting exemplars that confirm the current model.

Error reduction may be the obvious goal, but there are alternate criteria that may approximate it, with a more reasonable (though still high) computational demand. Model change (Settles et al., 2008) selects exemplars that are likely to cause a large change for the clas-

sification models parameters. In principal information resulting in a large parameter change is of greater value than minor tweaks, though it very much depends on the meaning of the parameters. Variance reduction (Cohn et al., 1994) selects exemplars to reduce the variance of the model, which can be interpreted as making the model more certain in its answers - it is related to QBC in this respect. Cohn et al. (1994) applied this to regression. The concept of a version space was introduced by Mitchell (1982) - it is defined as the set of model parameters that correctly classify the currently labelled data. Tong and Koller (2000) introduce a margin-based active learning method for SVMs. Selection from the pool is driven by reducing the size of the version space as quickly as possible, to find the best model in the least number of queries. Whilst the concept is sound their implementation requires that the data be separable, which is fatal in many real world scenarios.

## 2.4 Discovery

Most existing active learning studies assume that all classes are known a priori. Hodge and Austin (2004) give the likelihood criterion, which proceeds by querying the exemplars that have the lowest probability according to the classifier’s current model. Whilst often considered to be an uncertainty criterion it is better suited to finding new classes than refining the boundaries of existing classes, hence its inclusion here. Likelihood is limited by its inability to distinguish new classes from outliers, and to find classes that are inseparable from already detected classes.

Recently there have been a number of works that explicitly focus on the rare class discovery problem. Pelleg and Moore (2004) use an EM classifier with Gaussian distributions and adopt a variant of the likelihood criterion. Whilst it is specifically for finding rare classes the total number of rare classes must be provided up front to set the number of EM clusters. The model and active learning method are also inseparable. He and Carbonell (2007) perform density estimation and query exemplars based on identifying local maxima in the density using gradients. Like Pelleg and Moore (2004) this requires knowing how many unknown classes exist; additionally it also needs an estimate of how many exemplars belong to each class - for real problems this is unreasonable. Vatturi and Wong (2009) also take a density estimation approach. Mean shift at multiple scales is used to cluster the examples in the pool, and for each cluster the example nearest to the centre is selected to be queried. It gives strong performance for rare class discovery. However, both Vatturi and Wong (2009) and He

and Carbonell (2007) avoid interacting with the classification model. This is advantageous as it applies no restriction on the model, but problematic as it can only work to find new classes, not to improve the classification model, so poor results are expected for classification. Our approach can also work with any classification method, its only restriction being a requirement that the model provide probabilistic answers. It interacts with the classifier, and can hence work to improve classification performance.

## 2.5 Discovery and boundary refinement

This is the approach to which the presented (Haines and Xiang, 2011) belongs - where the discovery of new classes and refinement of the class boundaries for known classes are considered within a single framework. Stokes et al. (2008) work on network intrusion detection, where they treat the two goals separately. Batches of exemplars to label are provided, where some members have been selected based on uncertainty, and some have been selected due to being outliers; the ratio between the types is fixed. This does not work very well - at the start of training class discovery provides the greatest value, but as the process runs and all the classes are found it needs to focus on class boundary refinement. Hospedales et al. (2011) resolves this issue by heuristically selecting which approach to use. The two approaches are a generative classifier (kernel density estimate) with likelihood based selection, and a discriminative model (support vector machine), with selection based on uncertainty. As the querying progresses it switches between the models based on their past performance. This initially means it mostly uses the generative model to find new classes, but latter tends to be discriminative, to refine the boundary between classes. This is ideal as generative tends to work best for classification at the start, when there are few labelled exemplars, whilst discriminative models are ultimately better, but only when given enough data. Not surprisingly, it outperforms previous active learning methods which are designed for solving either class discovery or classification, not both. However, the method is entirely heuristic and includes parameters that need to be tuned for each scenario. Our approach shows similar behaviour when it comes to transitioning between discovery and refinement, but this behaviour is induced by the Dirichlet process assumption, without the need for heuristics.

## 2.6 Variations

We present pool based learning. This consists of having a pool of exemplars from which to choose the next one to be labelled. An alternate scenario is stream based active learning (Cohn et al., 1994) - in this case exemplars arrive continuously, and are not stored, so an instantaneous decision is required for each on if it should be given to the domain expert or not.

Active learning is traditionally applied to classification, as we do, but can also be applied to regression (MacKay, 1992). MacKay (1992) is actually concerned with experiment design, where one selects the most informative scientific experiments to run with limited time/budget, an area closely related to active learning. Transfer learning is related in some situations, by virtue of handling the relationship between known and unknown classes. An example of this is Lee and Grauman (2010), which uses the relationship between known classes and unknown classes to automatically infer the unknown classes, ready for human verification followed by further learning. Reinforcement learning (Kaelbling et al., 1996) is also closely related to the presented kind of active learning, via the *exploration-exploitation* problem.

## 3 Method

Given a pool of unlabelled instances the algorithm consists of a loop containing three steps,

1. A specific exemplar is selected from the pool.
2. It is labelled by the domain expert.
3. The model is updated with the new labelled exemplar. A previously unseen label will result in the creation of a new class in the model.

Our approach is responsible for the first step, and is itself broken down into three tasks,

1. For each exemplar a distribution over which class it belongs to is estimated, using the current model. It uses the Dirichlet process assumption to also calculate the probability of it belonging to an unknown class.
2. The probability of misclassification is calculated for each exemplar, which includes the possibility of it being misclassified due to it belonging to a new, unknown class.
3. An exemplar is selected, based on the misclassification probability.

These three tasks are detailed in the following subsections. Additionally a discussion of when to stop and a demonstration of the algorithms behaviour are also provided.

### 3.1 New class probability

Calculating the probability that an instance comes from an unknown class is problematic, as, by definition, nothing is currently known about the unknown classes. To resolve this it is assumed that a generative model of the data can be used, specifically a *Dirichlet process* (DP) mixture model. This is a valid assumption for most classification problems (Sethuraman, 1994).

A Dirichlet process (Ferguson, 1973) is typically used for non-parametric Bayesian models, e.g. density estimates (Escobar and West, 1995) and topic models (Teh et al., 2006). For the purpose of active learning however two properties are important: that it has clustering behaviour (Teh and Jordan, 2010), such that it expects the instances to be grouped into discrete classes; and that it considers an infinite number of classes, and hence dynamically adjusts the number of classes given the data.

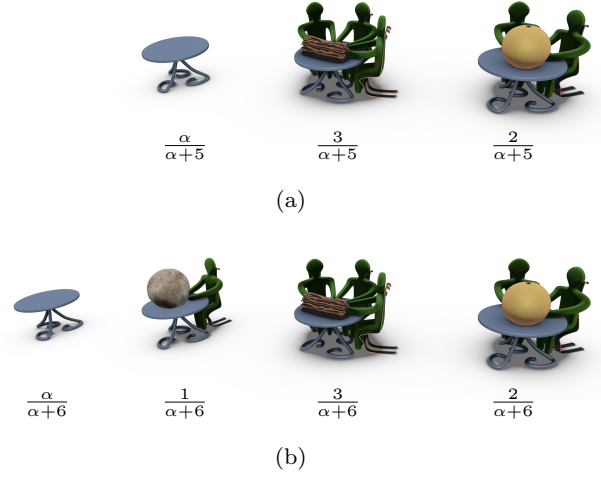
The Dirichlet process may be denoted as  $DP(\alpha, G_0)$ , where  $\alpha$  is its *concentration parameter* and  $G_0$  is its *base measure*. A draw from a DP provides a probability distribution over draws from the base measure, in the form of a Dirichlet distribution with infinitely many members. The clustering behaviour occurs because, even if the base measure is continuous, draws from it can be repeated when you draw from the infinite Dirichlet distribution. The stick-breaking process (Sethuraman, 1994) models this behaviour explicitly. Intuitively, we start with a stick of length 1, representing the entire probability mass, and keep breaking it in two. Each time we break it we get two parts - one is assigned to a draw from the base measure, and its length is the probability of drawing that particular item from the distribution we are generating, whilst the second part goes on to the next break. Specifically, this process generates an infinitely large set of sticks, of length  $s_i, i \in \{0, 1, \dots\}$ ,

$$s_i = s'_i \prod_{j=1}^{i-1} (1 - s'_j) \quad (1)$$

where  $s'_i$  can be thought of as the breaking ratio at each step. It is drawn from a beta distribution, which is dependent on the concentration parameter

$$s'_i \sim \beta(1, \alpha) \quad (2)$$

The concentration parameter therefore influences how much of each break goes to the current stick and how much is shared by future sticks. In the context of a mixture model, where each stick represents a cluster, a low value means most of the probability mass will be within a few clusters, whilst a high value means it is shared by many.



**Fig. 1** A representation of the Chinese restaurant process - customers sitting around tables with the chosen menu item in the centre. Under each table the probability of an arriving customer choosing it is given. If a person were to arrive at (a) and sit at the empty table the next state would be (b).

For the purpose of active learning all we need to calculate is the marginal posterior, i.e. given

$$G \sim DP(\alpha, G_0) \quad (3)$$

$$c_i \sim G \quad (4)$$

where  $c_i$  is the class of an exemplar it is defined as

$$P(c_i | \{c_j; j \in J_i\}, \alpha, G_0) \propto \quad (5)$$

$$\int \prod_{j \in J} P(c_j | G) P(G | \alpha, G_0) dG \quad (6)$$

$$J_i = J \setminus \{i\} \quad (7)$$

$$J = \{1, \dots, n\} \quad (8)$$

where  $n$  is the number of labelled exemplars. It is given by the Chinese restaurant process (Blackwell and MacQueen, 1973). The Chinese restaurant process is an analogy consisting of a restaurant containing an infinite number of tables at which customers sit. Each table represents a cluster in the mixture model, whilst the customers represent exemplars from the pool. On each table only a single dish is served, representing a single choice from the menu. This represents a draw from the base measure, and for active learning is the class associated with the cluster. When new customers arrive they either sit at a table with existing patrons, and consume the dish already assigned to the table, or they choose a previously unused table, for which a new dish is selected from the menu. These correspond to the instance belonging to an existing class and a new class, respectively. Each of the in-use tables is chosen proportional to the number of customers already sitting

at them, whilst a new table is chosen proportional to the concentration parameter. Note that whilst an infinite number of tables theoretically exist, corresponding to the components of the infinitely sized Dirichlet distribution, only used tables need to be tracked, making this a finite construction. This is illustrated in figure 1.

For each instance in the pool of unlabelled instances the aim is to compute the probability of it belonging to each existing class, and of it belonging to a new class, conditional on all previous instances for which the domain expert has provided a label. This is assuming a mixture-like model, where each table in the DP corresponds with a class assignment. Note that this is not a requirement for the classification model to also be a mixture model; it can be any model where  $P_c(\text{data}|\text{class})$  can be calculated. For the moment the existence of a prior,  $P(\text{data})$ , is also assumed, such that  $P_c(\text{data}|\text{class})$  is its posterior, using Bayes rule. Accordingly, the probability distribution for an instance is given as

$$P_n(c \in C \cup \{\text{new}\} | d) \propto \begin{cases} \frac{m_c}{\sum_{k \in C} m_k + \alpha} P_c(d|c) & \text{if } c \in C \\ \frac{\alpha}{\sum_{k \in C} m_k + \alpha} P(d) & \text{if } c = \text{new} \end{cases} \quad (9)$$

where  $d$  is the data for the considered instance,  $C$  is the set of known classes,  $m_c$  the number of instances labelled with class  $c$  and  $\alpha$  is the concentration parameter for the DP. Once normalised this provides a distribution for each instance that consists of the probability of the instance belonging to each of the known classes as well as to an unknown class. Two issues remain - how to set the concentration parameter and how to set the prior,  $P(\text{data})$ .

Instead of treating  $\alpha$  as a user set parameter a prior may be applied and Gibbs sampling used to estimate it, using the technique of Escobar and West (1995). The prior on  $\alpha$  is a gamma distribution,  $G(a, b)$ . This method proceeds by first sampling a quantity  $\eta$  given the current concentration, and then resampling the concentration given  $\eta$ .  $\eta$  given the concentration,  $\alpha$ , is given in terms of the beta distribution,  $\beta(.,.)$

$$\eta | \alpha, k, n \sim \beta(\alpha + 1, n) \quad (10)$$

where  $k$  is the number of classes that currently exist and  $n$  the number of examples distributed over the classes.  $\alpha$  given  $\eta$  is then a mixture of two gamma distributions,  $\Gamma(.,.)$

$$\alpha | \eta, k, n \sim \pi \Gamma(a + k, b - \log(\eta)) + (1 - \pi) \Gamma(a + k - 1, b - \log(\eta)) \quad (11)$$

where the ratio of the mixing terms is given by

$$\frac{\pi}{1 - \pi} = \frac{a + k - 1}{n(b - \log(\eta))} \quad (12)$$

Given a prior the mean of a number of Gibbs samples is used, after a burn in period Geman and Geman (1984). In this work a weakly-informative prior of  $\Gamma(1, 1)$  is used, with 128 samples used for both burn in and sampling the mean; for initialisation the concentration of the previous query is used.

A prior,  $P(\text{data})$ , is also required. Whilst a proper prior can certainly be used this term obviously parallels active learning methods based on density estimation (Such as Vatturi and Wong (2009)) - it defines how likely a sample is something useful, rather than an outlier. It follows that the prior must be selected based on the data in the pool, for which it is in effect going to be a density estimate. Given that real priors are often very simple, e.g. conjugate, a good density estimate will be beneficial and, as there is no reason to use an actual prior, a proper density estimate based on the initial pool is preferred.

### 3.2 Misclassification probability

Given the class membership probabilities,  $P_n(.)$ , which include the probability of belonging to a new class, an actual selection from the pool is required. The goal is to balance finding new classes against refining existing classes. A common approach to improving the existing model is to select instances that have a high degree of uncertainty in their classification given the current model. The most popular method is the entropy method, but entropy cannot be applied when there is a probability of an unknown class, at least not without the introduction of a free parameter. Several alternative approaches to entropy exist (Settles, 2009). One such approach is to calculate the probability of classifying an instance incorrectly. For instance this approach was implicitly used by Lewis and Gale (1994) for the purpose of text classification. They described it in terms of selecting instances with class probabilities that are closest to 0.5, which is equivalent. To include the possibility of a new class this idea has to be considered explicitly, and proper consideration of multiple classes has to be made.

Two assumptions are made - firstly that the classifier will select the class to which it has assigned the highest probability, noting that this only includes known classes, and secondly that the calculated distribution is an accurate estimate of what the true class of the instance could be, noting that it includes the possibility of a new class. It is then a simple matter to calculate

the probability of incorrectly classifying an instance,

$$P(\text{wrong}|d) = 1 - P_n(c'|d) \quad (13)$$

$$c' = \underset{c \in C}{\operatorname{argmax}} P_c(c|d) \quad (14)$$

where  $P_n(c|d)$  is the probability of the instance belonging to the selected class as calculated above, whilst  $P_c(c|d)$  is the probability calculated by the classifier, typically using Bayes rule with a  $P(c)$  term. If  $P(c)$  weights classes by the number of instances seen then  $P_n(c|d)$  and  $P_c(c|d)$  will be equivalent, other than  $P_c$  excluding the probability of a new class and hence being normalised differently. Alternatively, if a different prior on class probability is assumed, e.g. a uniform distribution, then this will not be the case.

It is important to note that the proposed misclassification probability (denoted as  $P(\text{wrong})$  hereafter) based active learning criterion is different from a conventional uncertainty criterion that focuses only on boundary refinement for existing classes. This is because  $P_n(c|d)$  includes the probability of the instance belonging to a new class (denoted as  $P(\text{new})$ ), which the classifier can never select. If the  $P(\text{new})$  value is high, the  $P(\text{wrong})$  value will also be high; similarly if the  $P(\text{new})$  value is low but the classifier is uncertain, so that none of the class probabilities are high, a high  $P(\text{wrong})$  will again be generated. Therefore the value of  $P(\text{wrong})$  is determined by two factors: the likelihood that the instance belongs to an unknown class, and how uncertain the current classifier is about the instance. Which of these two dominates is driven by the concentration parameter. Specifically, when it is high relative to the number of labelled instances selection becomes equivalent to using  $P(\text{new})$  directly, but as it heads to zero only classification uncertainty is considered, and the boundaries are refined. In practice the concentration parameter relative to the instance count tends to start high and drop to a low but constant level as the number of queries increases, i.e. as expected it starts by finding new classes, but as it sees more data and stops finding them it refocuses on boundary refinement.

### 3.3 Selection

Given the calculation of  $P(\text{wrong})$  for every item in the pool a specific item still needs to be selected, so it can be labelled. The obvious solution is to select the exemplar with the highest value of  $P(\text{wrong})$ , as the most useful. Experiments show this to be suboptimal - instead a *soft* selection strategy is used, where a random selection from the pool is made, weighted by  $P(\text{wrong})$ .

The choice is made because soft selection tends to provide better results, though it does depend on the problem. Hard selection is problematic because outliers often look like good candidates for new classes, and hence have a high  $P(\text{wrong})$  score, when they should probably be ignored. Soft selection avoids this issue as groups of samples with a moderate  $P(\text{wrong})$  score have a total weight larger than that of any given outlier, so the likelihood that a member of these groups will be chosen is greater than for the outliers. This can be seen as a density weighting, to avoid classifying exemplars that are irrelevant.

Implicit in this strategy is the assumption that the classifier assigns a single class to each exemplar, when the requirements of  $P(\text{wrong})$  require that it actually outputs a probabilistic assignment. It is relatively easy to make the conversion, but this actually compromises performance, because in testing a single class is assigned to each exemplar - unsurprising the approach works better if it uses the same assumptions made when testing it. A further assumption is made that the classifier will give an accurate estimate, which is clearly not true, especially when few queries have been made. However, if a Bayesian classifier is used it will express its uncertainty, and  $P(\text{wrong})$  will react accordingly, to improve that uncertainty and obtain a better classifier. Whilst non-Bayesian classifiers typically lack such a measure a suitable estimate of confidence can often be obtained.

### 3.4 Stopping Conditions

Active learning is concerned with limited resources - the fact that it takes time/money/energy to provide ground truth information for a classification algorithm. Eventually the querying has to stop. Three common options can be considered:

- **Query budget:** A fixed number of queries are performed.
- **Sufficient performance:** Enough queries are performed for classification performance to surpass a threshold. It can be estimated using  $n$  fold cross validation once enough queries have been performed to get an accurate enough estimate.
- **Cost-benefit analysis Dupuit (1952):** In many situations misclassification can have a directly attributed cost, as can providing further labelled exemplars - the total cost can then be minimised. To exemplify a widget factory may have a classifier to detect faulty products, alongside a given defect rate. The defect rate multiplied by the false negative rate of the classifier will give the percentage of faulty products sent to customers - multiply this by the

sales projections and the cost of handling a return and you obtain the money wasted by the classifiers mistakes. The false positive rate should also be factored in, in terms of throwing out usable widgets. Given the cost in employee time to train the classifier we can now work out at what point the cost of further training exceeds the value obtained (For a given product lifespan.), and hence when to stop training. Complex effects can exist, e.g. sending customers faulty products can generate bad publicity, making sales a function of the classifiers false probability rate.

The choice of scheme is scenario specific however, and as such we will not be exploring it further. However, by presenting results to a deep enough query count the above stopping conditions are implicitly represented using graphs of inlier rate against query count (Figure 6). Query budget is represented by seeing which is highest after a given number of queries (a vertical line), whilst performance is given by which algorithm crosses a given threshold first (a horizontal line). A cost benefit analysis is often represented by a straight line at an angle set by the relative costs of failure and further training. More sophisticated cost-benefit models can generate an arbitrary curve.

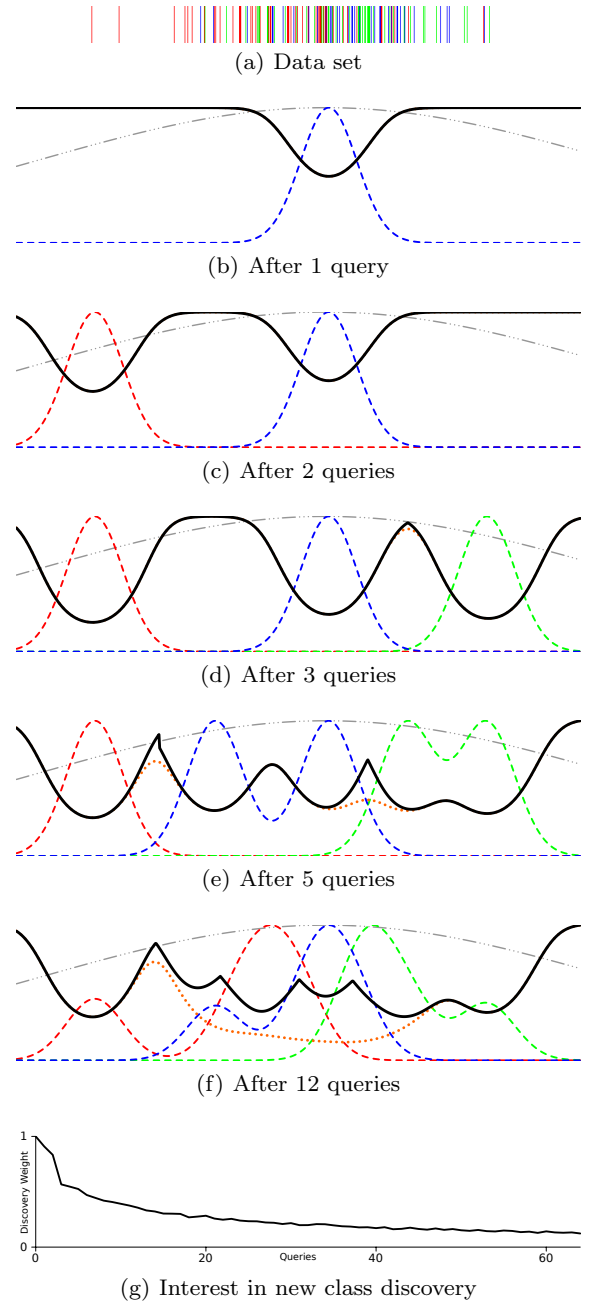
### 3.5 Demonstration

The presented approach,  $P(\text{wrong})$ , is now demonstrated and visualised using a 1D problem. Specifically, the 4D 3-class *iris* problem of Fisher (1936) is used, as obtained from the UCI repository (Frank and Asuncion, 2010). It is a classification problem where the task is to identify flower species based on flower shape measurements. Principal component analysis (PCA) is used to reduce the problem to a single dimension<sup>3</sup> - the resulting data set is visualised in Figure 2(a). Each line represents a member of the data set - horizontal position indicates the position projected to by PCA, whilst the three primary colours represent the three classes.

Many approaches can be selected for classification - for this and the other experiments the incremental kernel density estimation (KDE) method of Sillito and Fisher (2007) is used. It uses a Gaussian kernel whilst the number of mixture components is capped, to maintain a constant time incremental algorithm. When the cap is passed<sup>4</sup> mixture components are optimally merged,

<sup>3</sup> It is not really solvable after this, as the classes have a lot of overlap, but it is sufficient to illustrate the inner workings of the presented approach, whilst reducing it to 1D allows for a clean visualisation.

<sup>4</sup> We set this classifier parameter to 32.



**Fig. 2** 1D demonstration of the problem with 3 classes, with probability distributions. The prior is constant, indicated by the dot-dash grey line, whilst the distributions for the 3 classes use the 3 primary colours, dashed. Orange dots are used for the  $P(\text{new})$  metric, whilst  $P(\text{wrong})$  is given in black.

in terms of minimising the Kullback-Leibler divergence of the approximation. One density estimate over the pool is used as the pseudo-prior<sup>5</sup>, whilst each class also has a density estimate built from its members. A uniform prior over class assignment is used and kernel size

<sup>5</sup> A density estimate that we hallucinate is the prior for the classification algorithm.



is selected using leave one out cross-validation. Bayes rule is used to calculate the probability of belonging to each class, and the class with the highest probability selected as the answer.

Figures 2(b) through to 2(f) show the state of the system after the given number of queries. They are plots of probability values calculated for every 1D feature vector, where each plot has been normalised to fill the available height. Firstly, the prior is given by the grey line, and it remains constant as the algorithm runs. Each of the known classes is coloured using a primary colour. The  $P(\text{new})$  curve, giving the probability that a point on the line is going to belong to a new class, is given in orange whilst the  $P(\text{wrong})$  curve is given in black.  $P(\text{new})$  has been included as it makes the behaviour of  $P(\text{wrong})$  with regards to boundaries clearer. The  $P(\text{wrong})$  graph indicates how interested in a point the presented algorithm is, with the high points being the positions the algorithm is likely to select for its next query, conditional on such locations actually appearing in the data set. Figure 2(g) plots the sampled concentration normalised by the concentration plus the number of labelled instances, i.e. the weight assigned to new classes, given the number of queries made.

Firstly, a new class is found in each of the first 3 queries, and with the weight assigned to finding new classes dominating the two metrics have identical interests (the orange line is underneath the black line). After the third query a slight difference is evident in that  $P(\text{wrong})$  is more interested in examples that are on the boundary between the blue and green classes. The state after 5 queries demonstrates that, as the algorithm loses interest in finding new classes, as plotted in figure 2(g), the two approaches start to differ, with  $P(\text{wrong})$  showing greater interest in the classification boundaries whilst still maintaining an interest in areas where new classes could be. By 12 queries this is much more pronounced. This demonstration clearly shows the various behaviours expected - an interest in areas where either new classes could be or the boundary could be refined, with the latter gaining dominance as it loses interest in finding new classes. Figure 2(g) demonstrates how the level of interest in finding new classes drops as the algorithm makes more queries<sup>6</sup>.

## 4 Evaluation

Results are given for 13 data sets, consisting of the following classification problems:

- **glass**: Infer glass type given its chemical contents, for forensic investigation. Features include chemical properties and how it breaks.
- **ecoli**: Predict which part of a cell contains a protein localisation site, for E.coli.
- **segment**: Labelling regions from images of outdoor scenes, with labels such as grass, path and sky. Input is a small patch of pixels; output is the label for the centre pixel of the patch.
- **pageblocks**: Classifying regions from document scans, e.g. as text, picture or graphic. Features include colour ratios and measures of texture.
- **covertype**: Predicting forest cover type given geographic information, such as elevation and soil type.
- **thyroid**: Determining the disease that a thyroid has given observed and measured properties.
- **winequality**: Predict the quality of Portuguese wine given various chemical properties. Strictly speaking this is a quantised regression problem.
- **letters**: Recognising handwritten letters from the English alphabet. Input is images of each letter.
- **shuttle**: Infer the state of part of the space shuttles propulsion system, given various sensor readings, as relating to the Challenger disaster.
- **kdd99**: Data set used for the *3rd Knowledge Discovery and Data Mining Tools Competition* - uses a simulation of a military network with the goal being to detect intrusions given tcp dump data. The original data set included multinomial attributes, which have been concatenated as part of the feature vector, hence the high dimensionality of the problem.
- **gait**: Inferring the quantised walking direction from aligned silhouettes that have been averaged over multiple frames (Input is a greyscale image.), as in Han and Bhanu (2006). This data set was sampled to be unbalanced, such that each class is half the size of the next larger.
- **digits**: Recognising the handwritten digits, 0 – 9, given images of the digits. This data set was sampled to be unbalanced, such that each class is half the size of the next larger.
- **faces**: Large scale face recognition from images extracted using a face detector. Uses the preparation given by Guillaumin et al. (2009); additionally people who have less than 10 entries have been removed.

In all cases the original features provided by the data sets have been used, though some of the vision problems have been subjected to dimensionality reduction, by principal component analysis (To avoid using an entire image as input.). The last data set, *faces*, is an example of an extremely large scale problem, having over 300 classes. Various statistics summarising the prob-

<sup>6</sup> Note that concentration cannot be calculated until at least two classes have been found, hence the jump in the graph at that time.

Problem	Origin	Classes	Dimensions	Train	Test	Largest	Smallest	Queries
glass	Frank and Asuncion (2010)	6	10	107	107	34.58%	3.74%	107
ecoli	Frank and Asuncion (2010)	8	7	168	168	48.21%	0.60%	150
segment	Frank and Asuncion (2010)	7	18	318	317	47.80%	0.94%	150
pageblocks	Frank and Asuncion (2010)	5	10	3649	1824	89.28%	0.49%	150
covertypes	Frank and Asuncion (2010)	7	10	2500	2500	24.36%	3.56%	150
thyroid	Frank and Asuncion (2010)	3	21	3772	3428	92.47%	2.47%	150
winequality	Frank and Asuncion (2010)	6	11	2447	2446	45.24%	0.37%	150
letters	Frank and Asuncion (2010)	26	16	2620	6656	13.74%	0.31%	200
shuttle	Frank and Asuncion (2010)	7	9	10000	14500	77.72%	0.03%	150
kdd99	Frank and Asuncion (2010)	15	113	16825	16825	51.46%	0.04%	200
gait	Hospedales et al. (2011)	9	25	411	1942	48.66%	2.92%	150
digits	Hospedales et al. (2011)	10	25	8184	5000	50.05%	0.10%	200
faces	Huang et al. (2007)	330	32	5195	5195	10.95%	0.04%	1000

**Table 1** Details of the data sets used. Origin gives the source - most have come from the UCI repository. Classes is the number of classes in the classification problem, dimensions the length of the feature vector. Train gives the number of training exemplars, which is the size of the initial pool; test the number used for testing. Largest is the percentage of exemplars that belong to the largest class, smallest the percentage that belongs to the smallest class, to indicate how imbalanced the problem is. Queries is the number of queries performed - the values were chosen to match previous papers (Hospedales et al., 2011; Haines and Xiang, 2011; Loy et al., 2012), with the other data sets set consistently.

lems are given in table 1 - note that they vary in size, number of classes and class imbalance.

For testing we use the incremental KDE method of Sillito and Fisher (2007), as described in subsection 3.5 to classify, via Bayes rule, and also to provide a pseudo-prior. Testing consists of running through the active learning loop and using a separate test set to measure the balanced classification performance after each query. Different problems are run to different query depths, depending on the number of classes in the data set - see table 1. Multiple runs are performed<sup>7</sup>, to account for the variability with the stochastic algorithms, and the results averaged. Finally, two graphs may be plotted - either the balanced inlier rate<sup>8</sup> (classification) or the number of classes discovered (discovery) can be graphed against the number of queries. These graphs, for all data sets, may be found in figures 6 and 7. We also report the areas under these graphs, noting that strong classification performance is better aligned with the goal of training as good a classifier as possible with as few queries as possible.

Four algorithms are compared against the presented approach,  $P(\text{wrong})$ :

- **random**: Random selection - effectively a dumb algorithm that provides a baseline for performance.
- **entropy**: Calculates the entropy of the class distribution of each item in the pool, selecting the exem-

plar that requires the most information to encode draws from. This approach attempts to refine the boundary between classes.

- **likelihood**: Selects the exemplar from the pool that has the lowest probability of belonging to the current model - it selects outliers. This approach attempts to find new classes.
- **Hospedales et al. (2011)**: Balances class discovery and boundary refinement by selecting between one model for each, based on current model performance. A generative model (kernel density estimate.) with *likelihood* based selection is used for class discovery, whilst a discriminative model (svm) with *entropy* based selection is used for boundary refinement. Selection is entirely probabilistic, including the use of Gibbs functions on the active learning criteria. Results are only available for three of the datasets; it has the unfair advantages of having been tuned for them and using a better classifier.

Results are given by the graphs in Figure 6, with discovery graphs additionally given in Figure 7. Table 2 summarises the classification performance by giving the area under the graph; Table 3 does similarly for discovery performance. The classification graphs give the balanced inlier rate of the classifier trained after  $x$  queries, whilst the discovery graphs give the number of classes found after  $x$  queries. In both cases an average of many runs is presented. The presented approach is the best for 9 out of 14 problems. Its nearest competitor is *likelihood*, which wins the remaining 4. In the scenarios in which it misses first place  $P(\text{wrong})$  always comes second - it is *consistently* good. The same can not be said for likelihood, which on two occasions comes last. An approximate ordering by complexity has been applied

<sup>7</sup> 32 in all cases except for *kdd99* and *faces*, where it is 24 and 16 respectively due to their size.

<sup>8</sup> Balanced inlier rate is calculated as the average inlier rate for each class in the training set. Inlier rate is the number of correct classifications divided by the number of exemplars being classified. This can be interpreted as *recall* generalised to 3+ classes.

Problem	Random	Entropy	Likelihood	Hospedales et al. (2011)	$P(\text{wrong})$
glass	70.5 (3)	61.6 (4)	72.2 (2)		74.5 (1)
ecoli	83.4 (3)	66.9 (4)	90.0 (1)		84.4 (2)
segment	90.7 (3)	67.3 (4)	109.4 (1)		107.7 (2)
pageblocks	64.6 (4)	70.7 (3)	76.6 (2)		78.5 (1)
covertypes	71.2 (2)	50.9 (4)	62.4 (3)		72.6 (1)
thyroid	76.8 (2)	65.7 (4)	75.3 (3)		80.1 (1)
winequality	36.8 (3)	33.3 (4)	37.6 (1)		37.2 (2)
letters	59.8 (3)	11.1 (4)	75.8 (1)		66.9 (2)
shuttle	53.5 (4)	51.8 (5)	79.4 (2)	61.8 (3)	79.8 (1)
kdd99	92.1 (4)	96.1 (3)	96.9 (2)		146.6 (1)
gait	78.9 (3)	75.3 (4)	56.5 (5)	84.8 (2)	88.4 (1)
digits	54.6 (5)	57.1 (4)	61.9 (3)	69.5 (2)	69.7 (1)
faces	131.2 (2)	125.4 (3)	101.2 (4)		136.6 (1)
wins	0	0	4	0	9

**Table 2** Results, given as the area under the number of queries-inlier rate graph. The numbers in brackets give the positions. The results of Hospedales et al. (2011) have been included where available.

Problem	Random	Entropy	Likelihood	Hospedales et al. (2011)	$P(\text{wrong})$
glass	581.1 (3)	480.9 (4)	611.1 (1)		598.5 (2)
ecoli	946.6 (3)	793.8 (4)	1057.2 (1)		1011.7 (2)
segment	875.2 (3)	665.2 (4)	991.2 (1)		945.4 (2)
pageblocks	535.8 (4)	537.1 (3)	735.6 (1)		629.4 (2)
covertypes	983.8 (2)	825.8 (4)	978.2 (3)		993.2 (1)
thyroid	397.2 (4)	417.8 (2)	420.4 (1)		409.5 (3)
winequality	695.5 (3)	603.2 (4)	837.2 (1)		714.2 (2)
letters	3580.0 (3)	367.0 (4)	4443.4 (1)		3848.8 (2)
shuttle	486.2 (4)	423.5 (5)	950.5 (1)	933.2 (2)	923.4 (3)
kdd99	1490.6 (4)	1857.0 (3)	2418.0 (2)		2546.1 (1)
gait	1170.5 (5)	1183.8 (3)	1171.7 (4)	1253.1 (1)	1241.9 (2)
digits	915.2 (5)	974.0 (4)	1060.2 (3)	1207.4 (1)	1133.6 (2)
faces	183859.9 (3)	193598.0 (2)	179906.6 (4)		194776.3 (1)

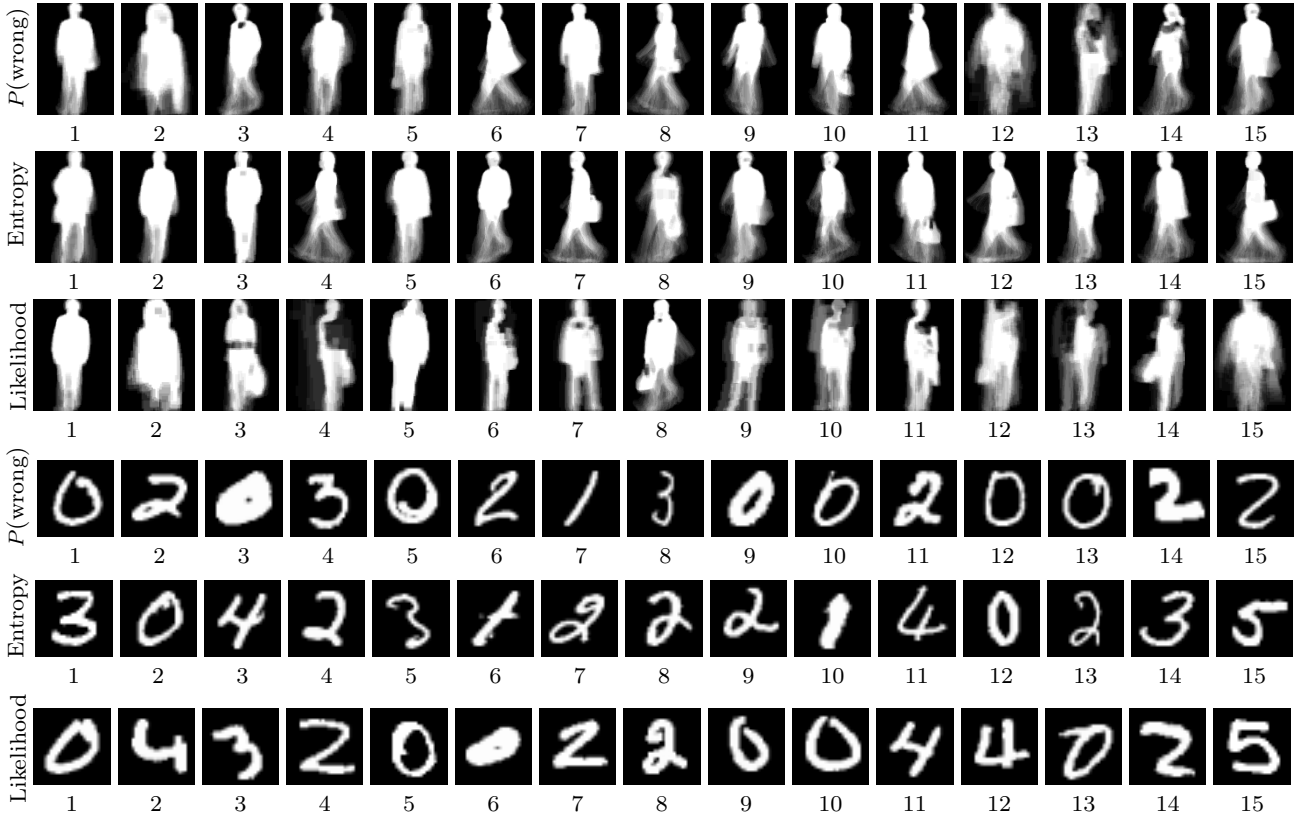
**Table 3** Results, given as the area under the number of queries-discovered classes graph. The numbers in brackets give the positions. The results of Hospedales et al. (2011) have been included where available.

to the problems, with the last few being scenarios where active learning is of the greatest value - for all of these examples  $P(\text{wrong})$  takes the lead. It wins for all of the vision problems except for *segment* and *letters*.

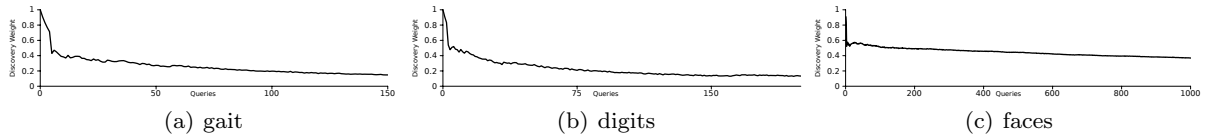
The *faces* data set is clearly the most challenging. It is a naturally imbalanced data set, where instances with less than 10 entries have been culled<sup>9</sup>. This chops off its thick tailed class size distribution - 47% of the data set has less than 20 items in the training set, meaning that a random selection has almost even odds of drawing from an approximately uniform set. As a result *random* does very well, as when it draws from a uniform distribution random selection has a good chance of finding a new class with every query. It only obtains second place however, with  $P(\text{wrong})$  in first, which is a strong result - *entropy* and *likelihood* both fail to even match *random*.

The initial selections of two vision-related data sets are visualised in Figure 3. Comparing the three approaches for the gait problem it is interesting to note the existence of gait energy images that have some kind of glitch, caused by a tracking failure or occlusion. The likelihood approach spends most of its time exploring these - because it is interested in outliers. Consequentially, it is wasting its queries on exemplars that are likely to confuse the classifier. Focusing on the *digits* problem note that  $P(\text{wrong})$  only explores 4 classes in the queries shown, the digits 0, 1, 2 and 3. Firstly, this data set has been geometrically distributed, with class '0' having the most entries, class '1' the second most, at half as many, and so on. So it starts by exploring the most common classes, which makes sense. Furthermore note that it queries '2' repeatedly, yet '1' only once, despite '1' being twice as common. This can be explained by the '1' class having relatively little variety when the '2' class has considerable variety, as demonstrated by the queries - each one is different from the others. In-

<sup>9</sup> Note that culling is for the entire data set, whilst separation into training and testing was purely random, so classes can have less than 10 entries in the pool.



**Fig. 3** The selected problem instances by one run of  $P(\text{wrong})$ , outlier and entropy, as indicated at the side, for the first 15 queries made. Top three rows show *gait*, bottom three show *digits* - the query number is beneath each image.



**Fig. 4** Plots of the inferred concentration value normalised by the concentration plus the number of instances that have already been labelled. It reflects how much effort  $P(\text{wrong})$  is putting in to finding new classes.

deed, the zeroes are subject to the greatest querying intensity, and show considerable variety.

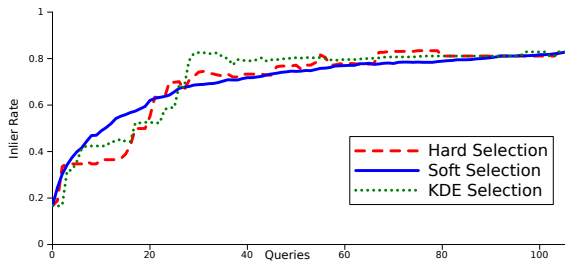
Figure 4 plots

$$\frac{\alpha}{\alpha + q} \quad (15)$$

against the number of queries made, where  $\alpha$  is the inferred concentration and  $q$  is the number of labelled exemplars, which is of course equal to the query count. This is effectively the weight assigned to discovering new classes. In all cases the expected happens - at the start it is very interested in discovering new classes, but as the number of queries progresses its interest drops and it focuses on refining the boundaries between the known classes. The *gait* and *digits* problems have a similar number of classes, and have similar profiles to their interest curve. In the case of *faces* however, which has dramatically more classes, the interest remains high for

all of the 1000 queries shown, as  $P(\text{wrong})$  is continuing to hunt for new classes.

Discovery performance is given in Figure 7, with the corresponding areas under the graphs in Table 3.  $P(\text{wrong})$  is not intended to optimise discovery. It instead attempts to obtain a good classifier with few queries, noting that discovering new classes will improve classification performance, as will improving the classification boundaries of existing classes - it effectively makes a trade off between these two actions, and in effect discovery performance is reduced so it can build a better classifier. The *likelihood* technique contrasts with this as it is only interested in discovery, so it is not surprising that *likelihood* gets the higher area the vast majority of the time. Indeed, a discovery focused algorithm such as Vatturi and Wong (2009) is expected to take the crown - for *shuttle* it has a winning score of



**Fig. 5** Comparison of different selection strategies. The graph shows the inlier rate of the test set on the  $y$  axis, as a function of the number of queries made, on the  $x$  axis, noting that the averages of many runs are shown. The area under the graphs is 74.266 for hard, 74.519 for soft and 75.891 for KDE.

970.5. For the last 3 problems the presented takes the lead. This certainly makes sense for *faces*, as the high class count means that a discovery-oriented approach is preferred, so the presented chooses to focus on discovery rather than boundary refinement.

To follow up subsection 3.3 Figure 5 demonstrates that soft selection is better than hard selection, using the *glass* problem. It also includes KDE selection - this involves reweighting the samples using a kernel density estimate (KDE) of the  $P(\text{wrong})$  weighted pool members (Using Gaussian kernels). This demonstrates the reasoning behind soft selection, by emulating it deterministically - this reweighting strategy in effect does approximately the same thing as soft selection<sup>10</sup>.

## 5 Conclusions

A state of the art active learning criterion has been presented and analysed, backed up by extensive results. It has all the properties that are desired for real world use:

- It both discovers unknown classes and refines the boundary between the known, automatically balancing these goals to maximise classification performance.
- The prior over concentration is the only parameter, and it does not need tuning - it was left as  $\Gamma(1, 1)$  throughout.
- Consistent top-tear performance on every problem tried. This and the above mean that it can be used without modification on many problems. Given the nature of active learning, where trying different ap-

<sup>10</sup> Whilst this strategy can always beat the presented approach it does so by introducing a scale parameter, which has to be selected for each problem. This is inappropriate, as doing multiple runs to find the best parameter obviates the entire purpose of active learning.

proaches mitigates its purpose, this confidence to treat  $P(\text{wrong})$  as a black box is essential.

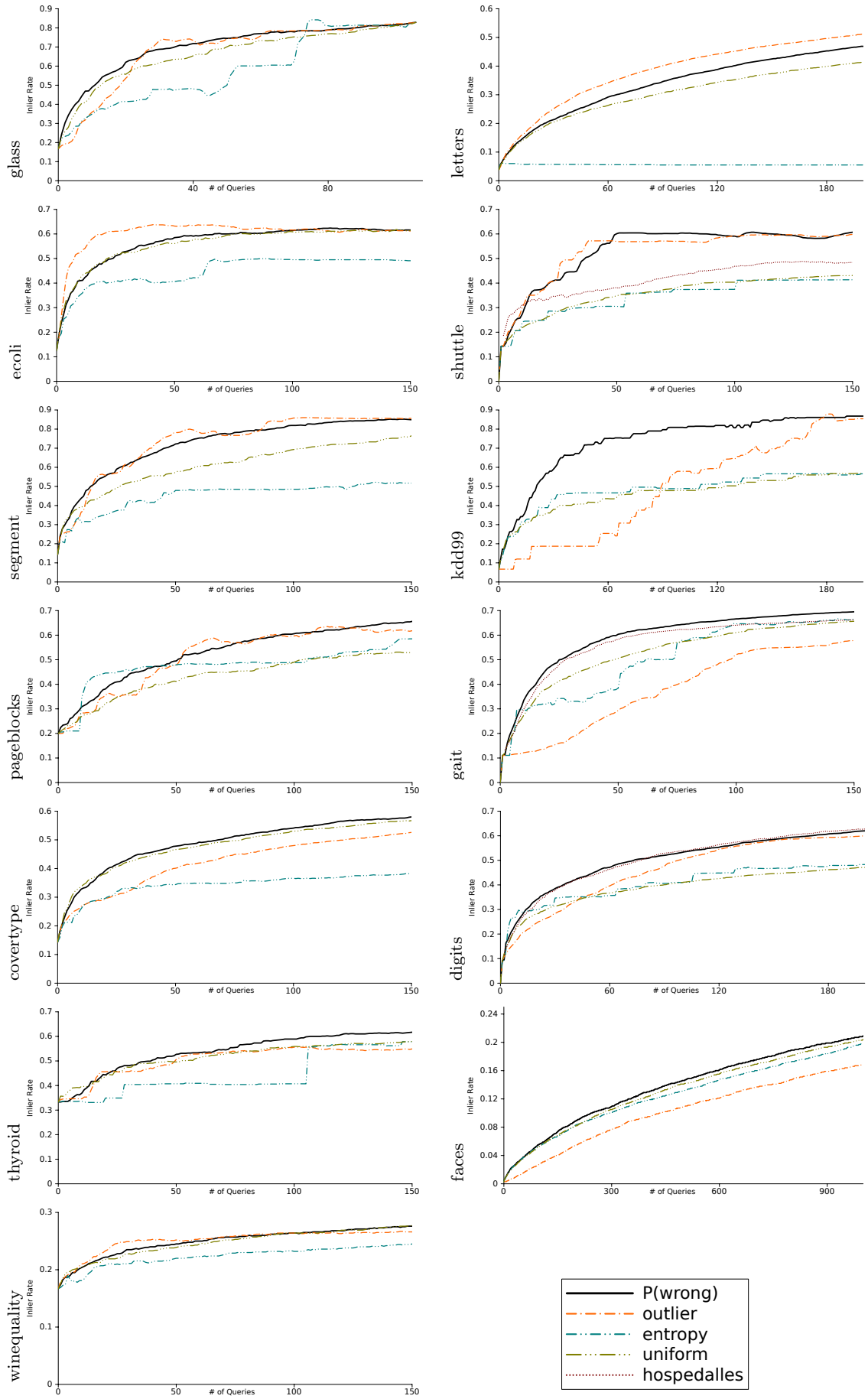
There are some limitations with  $P(\text{wrong})$ . Firstly, it is designed to work with unbalanced data - if run with balanced data it will continue to work, but random selection will typically do better. It aims to build as good a classifier as it can with the least number of queries - this is not the same as trying to discover one instance of every class. If discovery is the aim then better approaches exist, though there is some evidence that for certain problems, such as *faces*,  $P(\text{wrong})$  can do better than some discovery oriented approaches at discovery. Finally, the classifier needs to provide probabilistic information, though this is typically not an issue, as all generative approaches do so by definition, whilst discriminative approaches can provide it in some cases, e.g. random forests (Ho, 1995; Breiman, 2001), or be altered to provide it in others, e.g. support vector machines (Boser et al., 1992; Platt, 1999).

Future work could consider changes to the core algorithm, to improve performance. Alternative loss functions or a different QBC formulation for instance. This approach can be adjusted to other scenarios - Loy et al. (2012) already applied a variant of  $P(\text{wrong})$  to stream-based active learning, with a Pitman-Yor assumption instead of a Dirichlet process assumption and a QBC-like selection strategy. In demonstrating a (slight) advantage from using a Pitman-Yor process this suggests using more sophisticated priors that better match the expected structure of the data<sup>11</sup>. Whilst separation from the classification model is a definite advantage there are scenarios in which a tighter integration would prove beneficial. Variations designed for semi- or weak-supervision would be invaluable.

## References

- Abe, N. and Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. *ICML*, pages 1–9.
- Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2(4):319–342.
- Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Polya urn schemes. *Annals of Statistics*, 1(2):353–355.
- Boser, B. E., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Computational Learning Theory*, 5:144–152.

<sup>11</sup> There is even an interesting human interface issue of presenting such priors to a non-expert, such that they can communicate what they already know about a specific problem.



**Fig. 6** Graphs of inlier rate against number of queries, to present the classification performance of the algorithm.

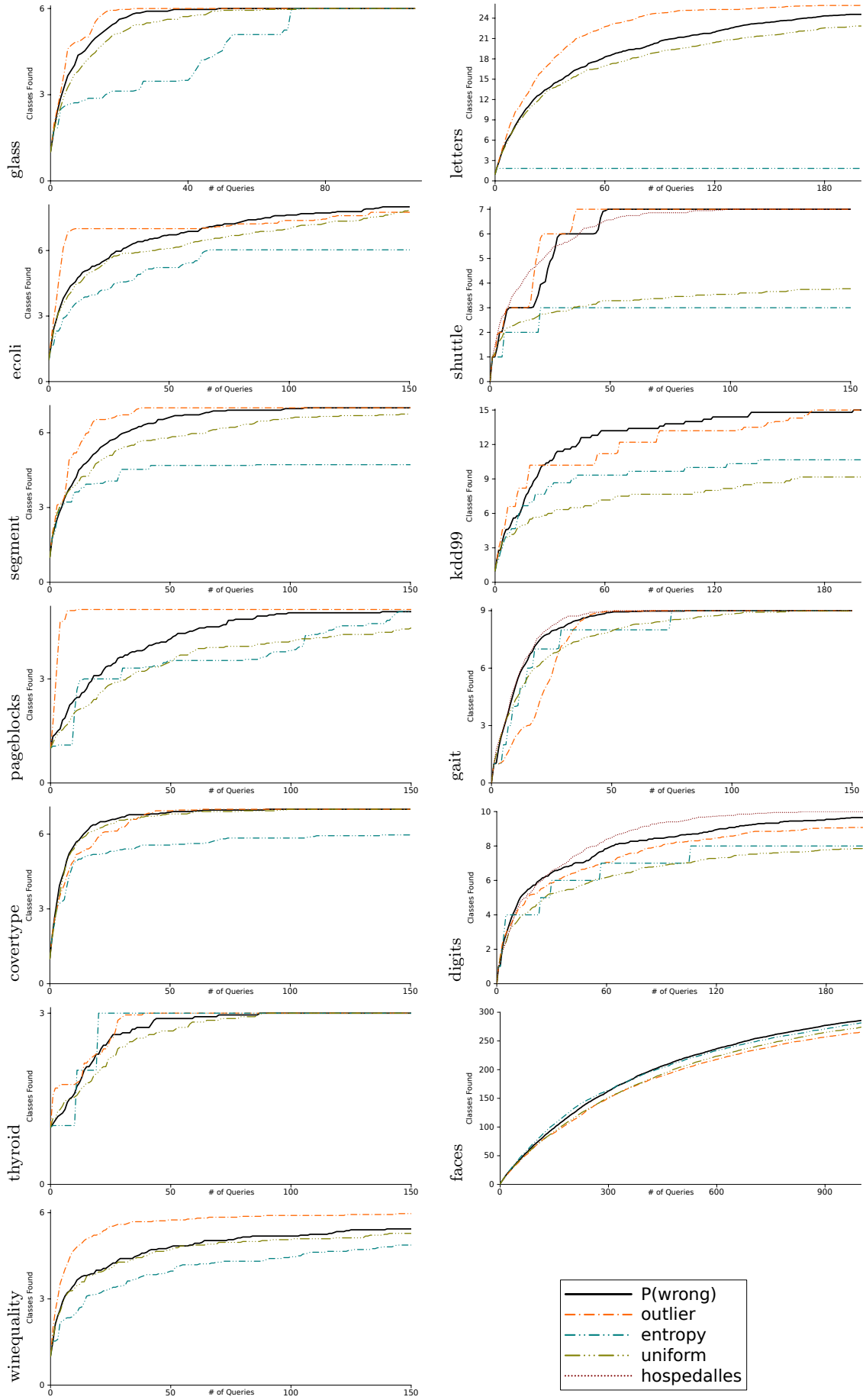


Fig. 7 Graphs of classes found against number of queries, to present the discovery performance of the algorithm.

- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.
- Culotta, A. and McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. *Proc. Nat. Conf. Artificial Intelligence*, pages 746–751.
- Dagan, I. and Engelson, S. (1995). Committee-based sampling for training probabilistic classifiers. *ICML*, pages 150–157.
- Dupuit, J. (1952). On the measurement of the utility of public works. *International Economic Papers*, 2:83–110.
- Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *J. American Statistical Association*, 90(430):577–588.
- Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.
- Frank, A. and Asuncion, A. (2010). UCI machine learning repository.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 6(6):721–741.
- Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you? metric learning approaches for face identification. *ICCV*.
- Haines, T. S. F. and Xiang, T. (2011). Active learning using dirichlet processes for rare class discovery and classification. *BMVC*.
- Han, J. and Bhanu, B. (2006). Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence*, 28(2):316–322.
- He, J. and Carbonell, J. G. (2007). Nearest-neighbor-based active learning for rare category detection. *Neural Information Processing Systems*, 21.
- Ho, T. K. (1995). Random decision forests. *Proc. Document Analysis and Recognition*, 1:278–282.
- Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126.
- Hospedales, T. M., Gong, S., and Xiang, T. (2011). Finding rare classes: Adapting generative and discriminative models in active learning. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 15.
- Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *J. Artificial Intelligence Research*, 4:237–285.
- Lee, Y. J. and Grauman, K. (2010). Object-graphs for context aware category discovery. *CVPR*, pages 1–8.
- Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proc. Conf. on Research and Development in Information Retrieval*, 17:3–12.
- Loy, C. C., Hospedales, T. M., Xiang, T., and Gong, S. (2012). Stream-based joint exploration-exploitation active learning. *CVPR*.
- MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604.
- McCallum, A. and Nigam, K. (1998). Employing em in pool-based active learning for text classification. *ICML*, pages 359–367.
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18:203–226.
- Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. Technical Report T2009:06, Swedish Institute of Computer Science.
- Pelleg, D. and Moore, A. (2004). Active learning for anomaly and rare-category detection. *Advances in Neural Information Processing Systems*, 17:1073–1080.
- Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, pages 61–74.
- Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. *ICML*, pages 441–448.
- Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.
- Settles, B. (2009). Active learning literature survey. Technical Report 1648, Uni. of Wisconsin-Madison.
- Settles, B., Craven, M., and Ray, S. (2008). Multiple-instance active learning. *NIPS*, 20:1289–1296.
- Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. *Proc. Workshop on Computational Learning Theory*, 5:287–294.
- Sillito, R. R. and Fisher, R. B. (2007). Incremental one-class learning with bounded computational complexity. *International Conference on Artificial Neural Networks*, 17:58–67.
- Stokes, J. W., Platt, J. C., Kravis, J., and Shilman, M. (2008). ALADIN: Active learning of anomalies to detect intrusion. Technical Report 2008-24, Microsoft



Research.

- Teh, Y. W. and Jordan, M. I. (2010). *Bayesian Non-parametrics*, chapter Hierarchical Bayesian Nonparametric Models with Applications. Cambridge University Press.
- Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *J. American Statistical Association*, 101(476):1566–1581.
- Tong, S. and Koller, D. (2000). Support vector machine active learning with applications to text classification. *ICML*, 2:45–66.
- Vatturi, P. and Wong, W.-K. (2009). Category detection using hierarchical mean shift. *Knowledge Discovery and Data mining*, 15:847–856.

each be satisfied, which is contrary to the goal. We therefore present  $P(\text{wrong})$  as formulated, as it is consistent - it never performs poorly, and usually gives top tier performance. Future work could consider inferring which data sets work best with different active learners.

## A Alternative choices

We now discuss some of the alternatives to the presented approach that were considered. Firstly, as discussed in subsection 3.3, one variant lead to an improvement, specifically soft selection over hard selection. Soft selection can be taken further - a parameter can be introduced as a power of the  $P(\text{wrong})$  value, to emphasis or de-emphasis large values. This can be tuned to get better results, but as a problem specific parameter it is of no value to active learning, as parameter tuning is incompatible with a single set of queries. The KDE variant in figure 5 is similar, except its parameter is fatally sensitive.

The probability of being wrong can be interpreted as an expectation over zero-one loss - alternative loss functions can be considered. Hinge loss for a multinomial distribution can be defined as the difference between the probability of the correct answer and the highest probability in the distribution, which is 0 if the correct answer has the greatest probability. It often undermined performance however.

Query by committee (QBC) was explored by Loy et al. (2012); however, their formulation really served as a probabilistic selection threshold function. Using multiple models it can be formulated to measure variance, so that  $P(\text{wrong})$  also focuses on areas with high model uncertainty<sup>12</sup>. Noting that there are two estimates - an estimate of what the actual class membership is, including the possibility of being something new, and an estimate of what the classifier is going to assign, we can use different models from a committee for these two roles. A QBC variant can then be defined using a committee where all assignment combinations are summed out, so a high QBC  $P(\text{wrong})$  score is obtained at boundaries between classes, in areas where new classes could be found, and where the current model has high uncertainty. This unfortunately resulted in too much emphasis being placed on boundary refinement.

For some problems the above variants are better. The issue is there is no way to predict *which* problems in advance, and for some problems they are much worse. Active learning is a scenario where you choose a method and apply it to your problem once - multiple runs require that the queries for

<sup>12</sup> With KDE this is obtained by training several classifiers on bootstrap samples from the training set. This achieves the goal of measuring model variance, but damages performance, so a fully trained version is kept to do actual classification.